

Table of Contents

This slide deck consists of slides used in 2 lecture videos in Week 1. Below is a list of shortcut hyperlinks for you to jump into specific sections.

- (page 2) [Week 1: Cookies and Sessions](#)
- (page 15) [Week 1:Django Sessions](#)

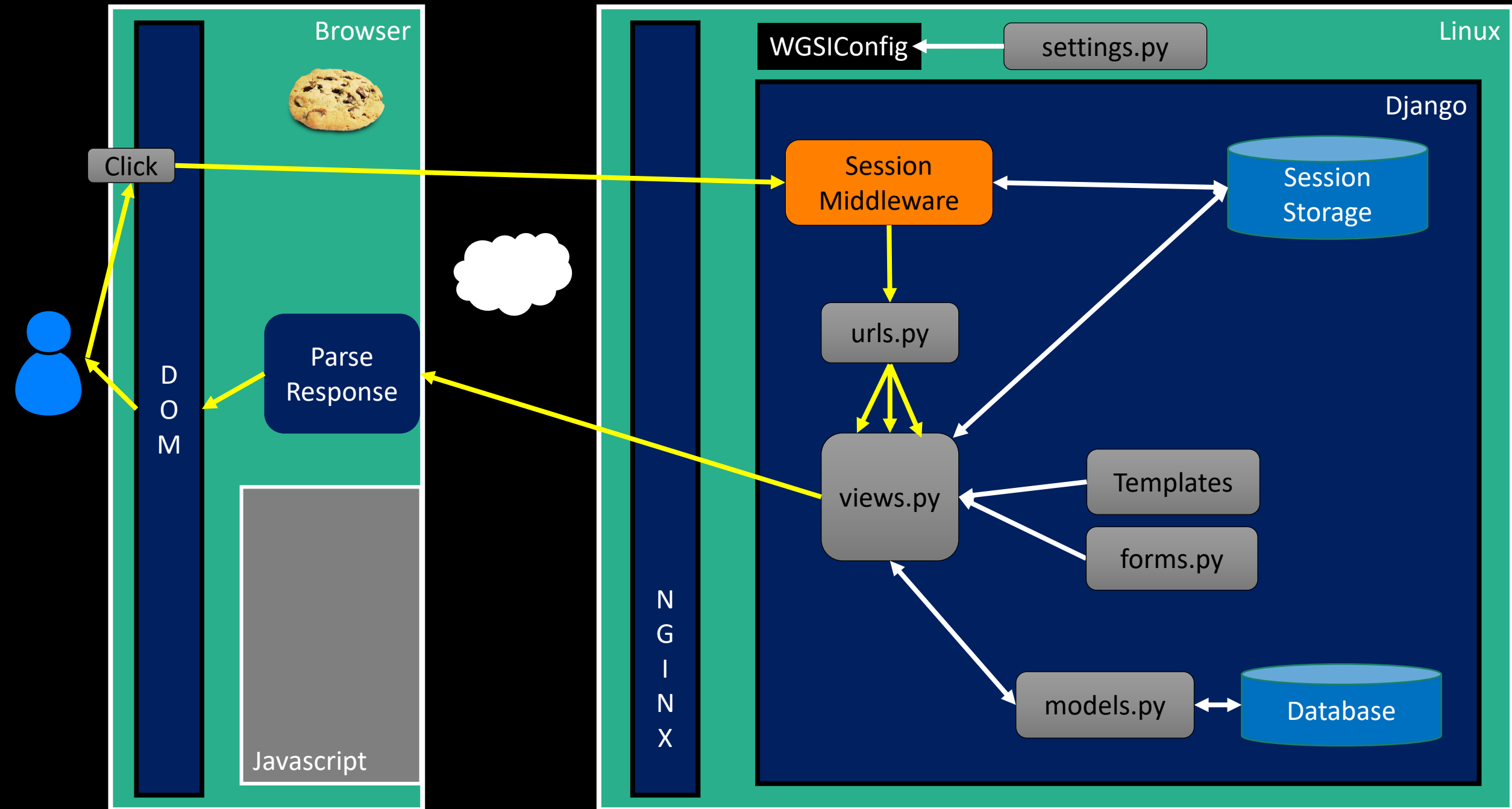
Charles Severance

www.dj4e.com

Cookies and Sessions

<https://samples.dj4e.com/session/>





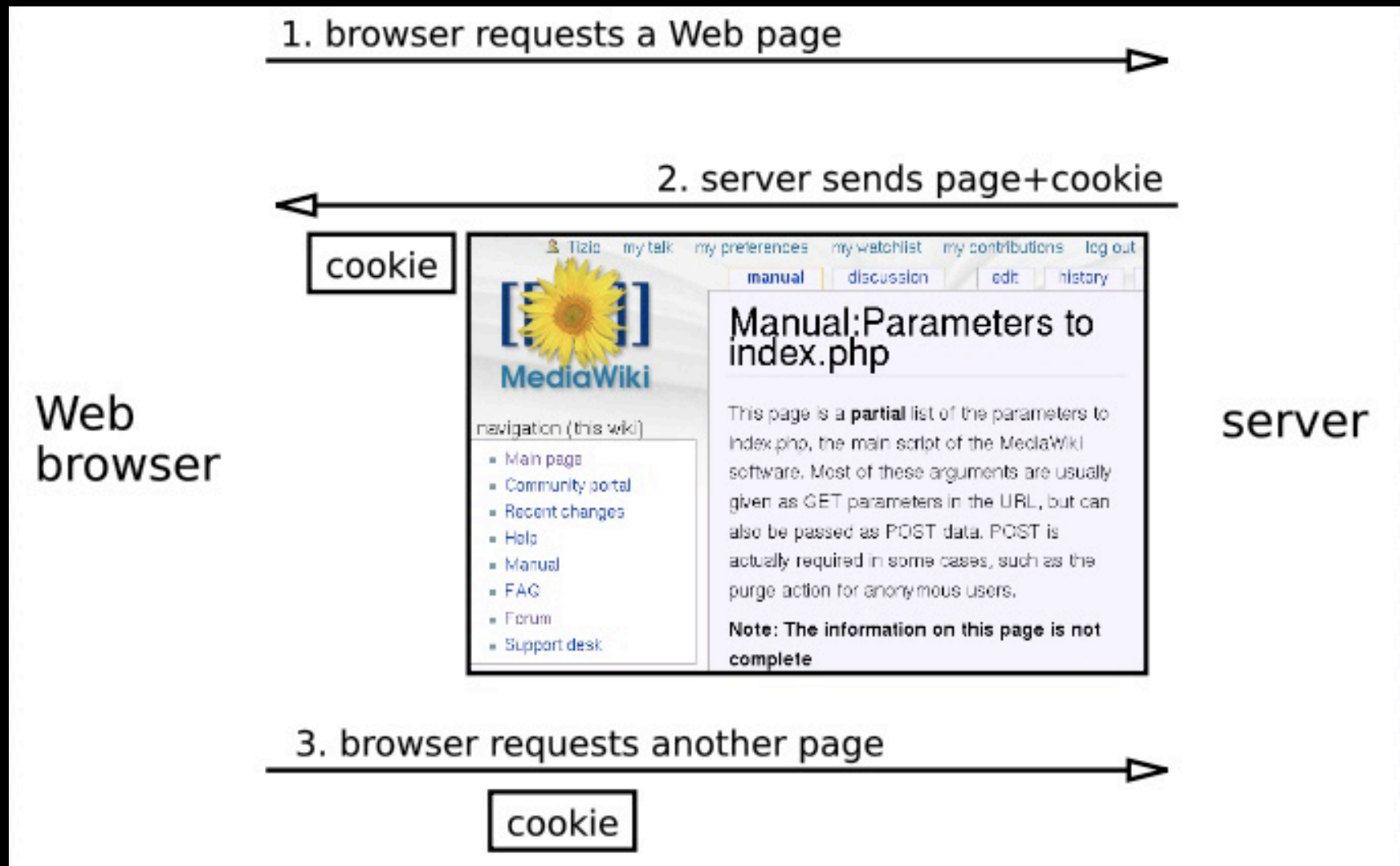
Multi-User / Multi-Browser

- When a server is interacting with many different browsers at the same time, the server needs to know *which* browser a particular request came from.
- Request / Response initially was stateless - all browsers looked identical . This was really bad and did not last very long at all.

Web Cookies to the Rescue

Technically, cookies are arbitrary pieces of data chosen by the Web server and sent to the browser. The browser returns them unchanged to the server, introducing a state (memory of previous events) into otherwise stateless HTTP transactions. Without cookies, each retrieval of a Web page or component of a Web page is an isolated event, mostly unrelated to all other views of the pages of the same site.

http://en.wikipedia.org/wiki/HTTP_cookie



http://en.wikipedia.org/wiki/HTTP_cookie

Cookies In the Browser

- Cookies are marked as to the web addresses they come from. The browser only sends back cookies that were originally set by the same web server.
- Cookies have an expiration date. Some last for years, others are short-term and go away as soon as the browser is closed

The screenshot shows a web browser window with the address bar displaying `https://samples.dj4e.com/session/`. The page content includes the text "This is the sessions sample code" and "This page is coming from a file in templates/session/main.html", followed by two links: [If you give a browser a cookie...](#) and [Fun with sessions](#).

The browser's developer tools are open, specifically the "Storage" tab. The left sidebar shows a tree view with "Cache Storage", "Cookies", "Indexed DB", "Local Storage", and "Session Storage". The "Cookies" section is expanded, showing a table of cookies for the domain `https://samples.dj4e.com`.

Name	Domain	Path	Expires	LastAccessed	Value	HttpOnly	SameSi...
<code>__cfduid</code>	<code>.dj4e.com</code>	<code>/</code>	Mon, 28 Sep 2020...	Sun, 29 Sep 2019 ...	<code>d15bb58c09d1...</code>	<code>true</code>	Unset

<https://samples.dj4e.com/session/>

<https://github.com/csev/dj4e-samples/blob/master/session/home/views.py>

<https://github.com/csev/dj4e-samples/blob/master/session/home/views.py>

```
def cookie(request):  
    print(request.COOKIES)  
    resp = HttpResponse('C is for cookie and that is good enough for me...')  
    resp.set_cookie('zap', 42) # No expired date = until browser close  
    resp.set_cookie('sakaicar', 42, max_age=1000) # seconds until expire  
    return resp
```

In the log:

```
[29/Sep 23:17:55] "GET /session/cookie HTTP/1.1" 200 26  
{'sessionid': 'xy414ikma0p80jw19mrg146inewhzp3s', 'zap':  
'42', 'sakaicar': '42'}
```

The screenshot shows a web browser window with the address bar displaying `https://samples.dj4e.com/session/`. The page content includes the text "This is the sessions sample code" and "This page is coming from a file in templates/session/main.html", followed by two links: [If you give a browser a cookie...](#) and [Fun with sessions](#).

The browser's developer tools are open to the "Storage" tab, showing a table of cookies for the domain `https://samples.dj4e.com`. The table has columns for Name, Domain, Path, Expires, LastAccessed, Value, HttpOnly, and SameSite.

Name	Domain	Path	Expires	LastAccessed	Value	HttpOnly	SameSite
<code>__cfduid</code>	<code>.dj4e.com</code>	<code>/</code>	Mon, 28 Sep 2020...	Sun, 29 Sep 2019 ...	<code>d15bb58c09d1...</code>	true	Unset

<https://samples.dj4e.com/session/>

<https://github.com/csev/dj4e-samples/blob/master/session/home/views.py>

4E samples.dj4e.com/session/ X +

4E samples.dj4e.com/session/cookie X +

https://samples.dj4e.com/session/cookie

Most Visited Drc SakaiCar Sakai Tsugi CRsera Teach LXP UMSI GMU GMD YT

This is the session

This page is controlled by cookies

C is for cookie and that is good enough for me...

- [If you give me a cookie](#)
- [Fun with cookies](#)

```
def cookie(request):  
    print(request.COOKIE)  
    resp = HttpResponse('C is for cookie and that is good enough for me...')  
    resp.set_cookie('zap', 42) # No expired date = until browser close  
    resp.set_cookie('sakaicar', 42, max_age=1000) # seconds until expire  
    return resp
```

Inspector Console Debugger Style Editor Performance Memory Network Storage

Filter URLs

Persist Logs Disable cache No throttling HAR

Status	Method	Domain	File
200	GET	samples.dj4e.com	cookie
200	GET	samples.dj4e.com	favicon.ico

Headers Cookies Params Response Timings Security

expect-ct: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"

server: cloudflare

set-cookie: zap=42; Path=

set-cookie: sakaicar=42; expires=Sun, 29 Sep 2014 03:03:03 GMT; Max-Age=1000; Path=

x-clacks-overhead: GNU Terry Pratchett

X-Firefox-Spdy: h2

2 requests 15.09 KB / 2.59 KB transferred Finish: 2.10 s

4E samples.dj4e.com/session/

4E samples.dj4e.com/session/cookie

4E samples.dj4e.com/session/cookie

← → ↺

Most Visited

This is the session

This page is coming from the cache

- If you give me a cookie
- Fun with cookies

Inspector

Cache Storage

Cookies

Indexed DB

Local Storage

Session Storage

200 GET

200 GET

2 requests

← → ↺

Most Visited

C is for cookie and that is good enough for me...

← → ↺

https://samples.dj4e.com/session/cookie

Search

Most Visited

Drc

SakaiCar

Sakai

Tsugi

CRsera

Teach

LXP

UMSI

GMU

GMD

YT

Inspector

Console

Debugger

Style Editor

Performance

Memory

Network

Storage

Filter items

Name	Domain	Path	Expires	LastAccessed	Value	HttpOnly	SameSite
__cfduid	.dj4e.com	/	Mon, 28 Sep 2020...	Sun, 29 Sep 2019 ...	d15bb58c09d...	true	Unset
sakaicar	samples.dj...	/	Sun, 29 Sep 2019 ...	Sun, 29 Sep 2019 ...	42	false	Unset
zap	samples.dj...	/	Session	Sun, 29 Sep 2019 ...	42	false	Unset

4E samples.dj4e.com/session/ X +

4E samples.dj4e.com/session/cookie X +

4E samples.dj4e.com/session/cookie X +

4E samples.dj4e.com/session/cookie X +

Most Visited

This is the session

This page is controlled by cookies

- [If you give me a cookie](#)
- [Fun with cookies](#)

C is for cookie and that is good enough for me...

Inspector

Cache Storage

Cookies

Indexed DB

Local Storage

Session Storage

200 GET

200 GET

2 requests

https://samples.dj4e.com/session/cookie

Most Visited Drc SakaiCar Sakai Tsugi CRsera Teach LXP UMSI GMU GMD YT

Inspector Console Debugger Style Editor Performance Memory Network Storage

Filter URLs

Persist Logs Disable cache No throttling HAR

Status	Method	Domain	File
200	GET	samples.dj4e.com	cookie
200	GET	samples.dj4e.com	favicon.ico
200	GET	samples.dj4e.com	cookie
200	GET	samples.dj4e.com	favicon.ico

4 requests 30.19 KB / 5.17 KB transferred Finish: 6.23 min

Headers Cookies Params Response Timings Stack Trace


- Cache-Control: no-cache
- Connection: keep-alive
- Cookie: __cfduid=d15bb58c09d1c97435a5f...69796218; zap=42; sakaicar=42
- Host: samples.dj4e.com
- Pragma: no-cache
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Macintosh; Intel... Gecko/20100101 Firefox/69.0

Time



Browser




Browser
zap=42 




zap=42



Browser
zap=42 

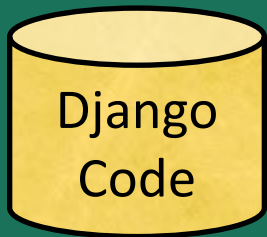


Browser
zap=42 



Apache

set_cookie() request.COOKIES request.COOKIES request.COOKIES



Django
Code

Django Sessions

<https://samples.dj4e.com/session/sessfun>

<https://github.com/csev/dj4e-samples/tree/master/session>



Click

D
O
M

Parse
Response

Javascript

Browser



N
G
I
N
X

WGSISConfig

settings.py

Linux

Django

Session
Middleware

urls.py

views.py

Templates

forms.py

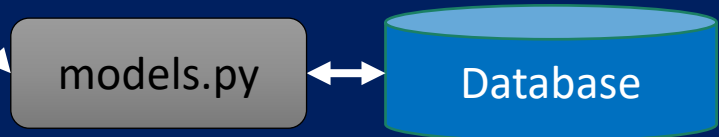
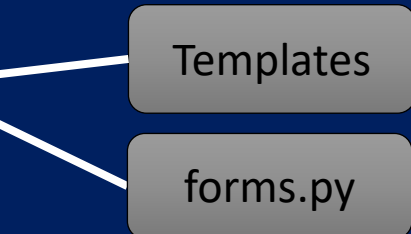
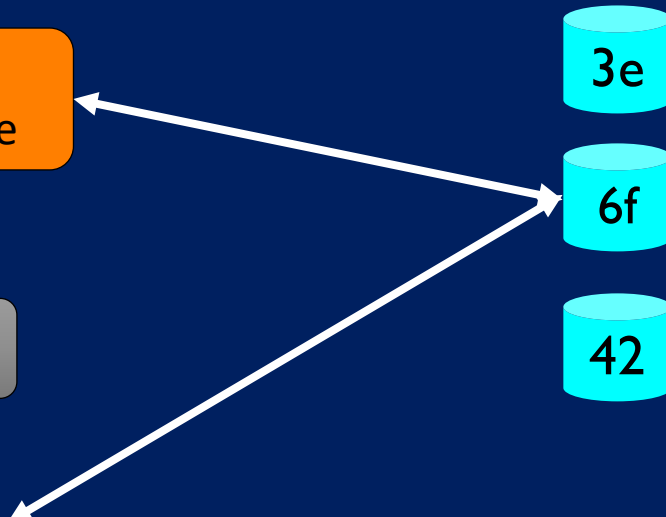
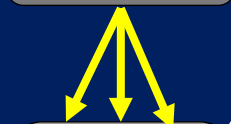
models.py

3e

6f

42

Database



In the Server - Sessions

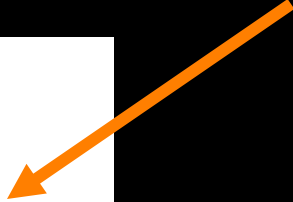
- In most server applications, as soon as we start a session for a new (unmarked) browser we create a session.
- We set a session cookie to be stored in the browser, which indicates the session id in use – gives this browser a unique “mark”.
- The creation and destruction of sessions is handled by a Django middleware that we use in our applications.

Session Identifier

- A large, random number that we place in a browser cookie the first time we encounter a browser
- This number is used to pick from the many sessions that the server has active at any one time.
- Server software stores data in the session that it wants to have from one request to another from the same browser.
- Shopping cart or login information is stored in the session in the server.

Middleware

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]
```



<https://github.com/csev/dj4e-samples/blob/master/dj4e-samples/settings.py>

Default – Store Sessions in the Database

```
$ python3 manage.py migrate
```

Operations to perform:

Apply all migrations: admin, auth, contenttypes, sessions

Running migrations:

Applying contenttypes.0001_initial... OK

Applying auth.0001_initial... OK

Applying admin.0001_initial... OK

Applying admin.0002_logentry_remove_auto_add... OK

...

Applying auth.0009_alter_user_last_name_max_length... OK

Applying sessions.0001_initial... OK





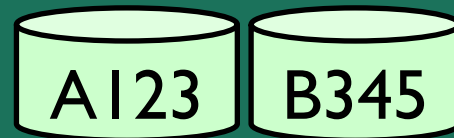
Browser
S=A123

Browser
S=B345

Browser
S=C678



Django Session Middleware



Web Server

Time



Browser
S=C123

Browser
S=C123

Browser
S=C123

Browser
S=C123



request
POST

session

request
GET

session

request
POST

session

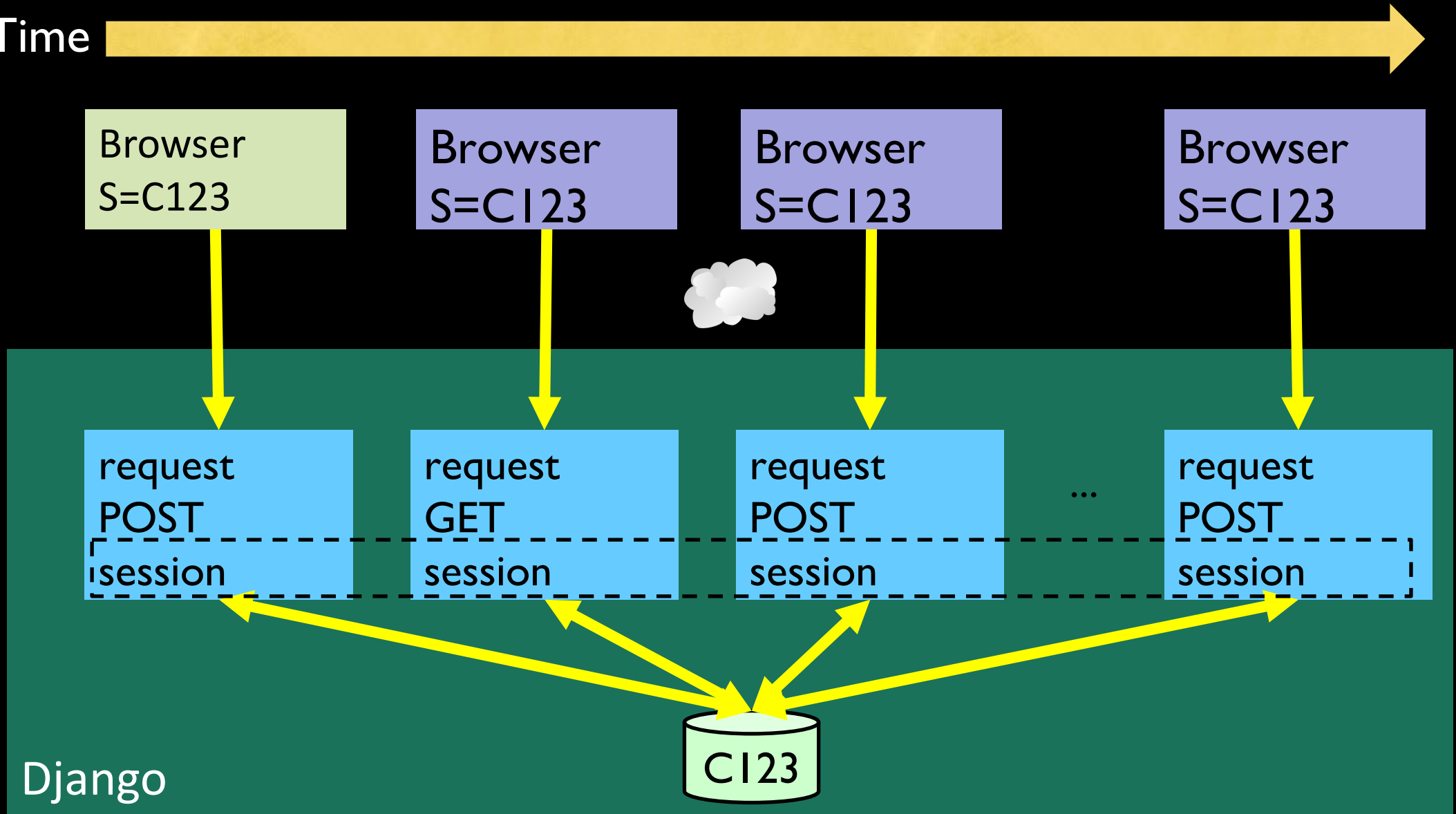
...

request
POST

session

Django

C123



Django Sessions

- The incoming `request` object has a `request.session` attribute that we can treat like a dictionary that persists from one request to the next request
- As long we have the session middleware enabled in `settings.py` and the database table, and the browser allows cookies, we just store and read `request.session` in our views and pretend it is "magic"

```
def sessfun(request) :  
    num_visits = request.session.get('num_visits', 0) + 1  
    request.session['num_visits'] = num_visits  
    if num_visits > 4 : del(request.session['num_visits'])  
    return HttpResponse('view count='+str(num_visits))
```

<https://github.com/csev/dj4e-samples/blob/master/session/views.py>

The screenshot shows a web browser window with the address bar displaying `https://samples.dj4e.com/session/`. The page content includes the text "This is the sessions sample code" and "This page is coming from a file in templates/session/main.html", followed by two bullet points:

- [If you give a browser a cookie...](#)
- [Fun with sessions](#)

The browser's developer tools are open to the "Storage" tab. On the left, the "Cookies" section is expanded, showing a table of cookies for the domain `https://samples.dj4e.com`. The table has columns for Name, Domain, Path, and Expires. One cookie is listed: `__cfduid` with domain `.dj4e.com` and path `/`. The right pane shows the details for this cookie, including its value `"d15bb58c09d...1569796218"` and various attributes like `CreationTime`, `Domain`, `Expires`, `HostOnly`, `HttpOnly`, `LastAccessed`, `Path`, and `SameSite`.

Name	Domain	Path	Expires
<code>__cfduid</code>	<code>.dj4e.com</code>	<code>/</code>	Mon, 28 Sep 2020...

Filter values:

- `__cfduid`: `"d15bb58c09d...1569796218"`
- `CreationTime`: `"Sun, 29 S...30:18 GMT"`
- `Domain`: `"dj4e.com"`
- `Expires`: `"Mon, 28 Sep ...22:30:18 GMT"`
- `HostOnly`: `false`
- `HttpOnly`: `true`
- `LastAccessed`: `"Sun, 29 S...0:46 GMT"`
- `Path`: `"/"`
- `SameSite`: `"Unset"`

`https://github.com/csev/dj4e-samples/blob/master/session/views.py`

4E samples.dj4e.com/session/

← → ↺ 🏠

Most Visited Drc

This is the sessions sam

This page is coming fro

- [If you give a bro](#)
- [Fun with session](#)

Inspector

Cache Storage

Cookies

https://samples.dj4e.com

Indexed DB

Local Storage

Session Storage

4E samples.dj4e.com/session/sess

← → ↺ 🏠

https://samples.dj4e.com/session/sessf

Search

Most Visited Drc SakaiCar Sakai Tsugi CRsera Teach LXP UMSI GMU GMD YT

view count=1

Inspector Console Debugger Style Editor Performance Memory Storage

Cache Storage

Cookies

https://samples.dj4e.com

Indexed DB

Local Storage

Session Storage

Filter items

Name	Domain	Path	Expires
__cfduid	.dj4e.com	/	Mon, 28 Sep 2020...
sessionid	samples.dj...	/	Sun, 13 Oct 2019 ...

Filter values

Data

sessionid: "gj2233cwnot...uw0u4f3h6u"

CreationTime: "Sun, 29 S...08:28 GMT"

Domain: "samples.dj4e.com"

Expires: "Sun, 13 Oct ...23:08:28 GMT"

HostOnly: true

HttpOnly: true

LastAccessed: "Sun, 29 S...8:28 GMT"

Path: "/"

SameSite: "Lax"

The image displays three overlapping browser windows, all showing the URL `samples.dj4e.com/session/`. The windows are arranged from left to right, with the rightmost window being the most prominent.

The rightmost window shows the page content: "view count=2". Below the page content, the browser's developer tools are open, specifically the Storage Inspector. The Storage Inspector shows a table of cookies and session storage data.

Name	Domain	Path	Expires
__cfduid	.dj4e.com	/	Mon, 28 Sep 2020...
sessionId	samples.dj...	/	Sun, 13 Oct 2019 ...

The Storage Inspector also shows the session storage data:

- sessionId:** "gj2233cwnot...uw0u4f3h6u"
- CreationTime:** "Sun, 29 S...08:28 GMT"
- Domain:** "samples.dj4e.com"
- Expires:** "Sun, 13 Oct ...23:08:28 GMT"
- HostOnly:** true
- HttpOnly:** true
- LastAccessed:** "Sun, 29 S...8:28 GMT"
- Path:** "/"
- SameSite:** "Lax"

```
$ sqlite3 db.sqlite3
```

```
SQLite version 3.24.0 2018-06-04 14:10:15
```

```
Enter ".help" for usage hints.
```

```
sqlite> .tables
```

auth_group	auth_user_user_permissions
auth_group_permissions	django_admin_log
auth_permission	django_content_type
auth_user	django_migrations
auth_user_groups	django_session

```
sqlite> .mode column
```

```
sqlite> select * from django_session;
```

```
vosaoain2dzw0o8bzlgsmovdbkp574us
```

```
YThiZWRjMjQ1NzZhMzYzMTBhZjYxNWI2ZDgyODI1Y2ExODI2MTJjNzp7Im51bV92aXNpdHMiOjF9  
2019-02-21 15:18:34.995362
```

```
vii016kh2vzqpm0uw3or4qrqxddmwisx
```

```
OWNkOGQxYjg4NzlkN2ZhOTc2NmU1ODY0NWMzMmQ4YjdhMzM4OTJhNjp7Im51bV92aXNpdHMiOjJ9  
2019-02-21 15:32:52.555061
```

```
sqlite> .quit
```

What is in the Django Session Table?

```
$ python3
>>> import base64
>>> x = base64.b64decode(
... 'OWNkOGQxYjg4NzlkN2ZhOTc2NmU1ODY0NWMzZmQ4YjdhMzM4OTJhNjp7Im51bV92aXNpdHMiOjJ9')
>>> print(x)
b'9cd8d1b8879d7fa9766e58645c3fd8b7a33892a6:{"num_visits":2}'
>>> import json
>>> data = json.loads(x[41:])
>>> print(data)
{'num_visits': 2}
>>>
```

Summary

- HTTP Cookies
- Sessions
- Using Sessions in Django

Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) as part of www.wa4e.com and made available under a Creative Commons Attribution 4.0 License. Please maintain this slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan
School of Information

Insert new Contributors and Translators here including names and dates

Continue new Contributors and Translators here

Additional Source Information

- Cookie Image: By brainloc on sxc.hu (Bob Smith) (stock.xchng) [CC BY 2.5 (<http://creativecommons.org/licenses/by/2.5>)], via Wikimedia Commons
- Portions of the text of these slides is adapted from the text www.djangoproject.org web site. Those slides which use text from that site have a reference to the original text on that site. Django is licensed under the three-clause BSD license.